

# The longest increasing subsequence of random separable permutations

Thomas Budzinski

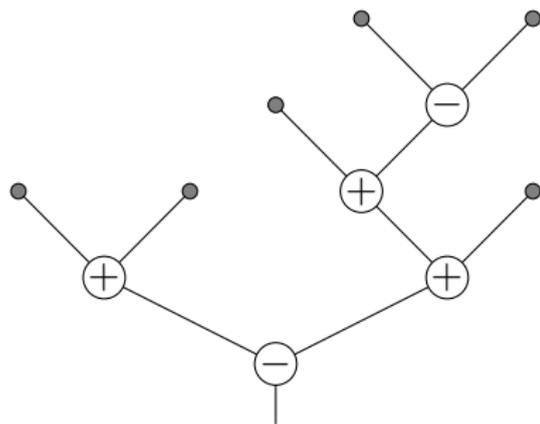
ENS de Lyon

2025, October 28th

Seminari di Probabilità, Analisi Stocastica e Statistica

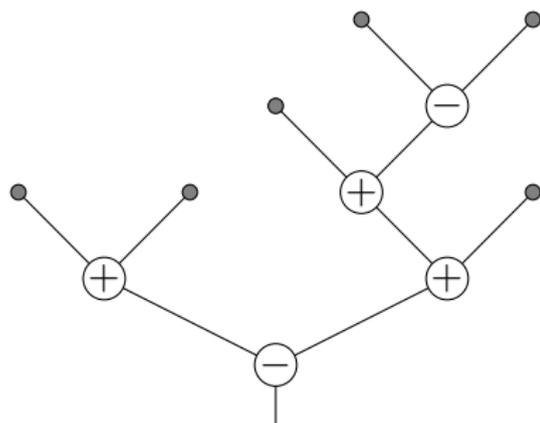
Joint work with Arka Adhikari, Jacopo Borga, William da Silva  
and Delphin Sénizergues

# Signed binary trees



- Trees with  $n$  leaves (0 child) and  $n - 1$  nodes (2 children),
- distinction left child/right child,
- nodes are decorated with  $+/-$  signs.

# Signed binary trees



- Trees with  $n$  leaves (0 child) and  $n - 1$  nodes (2 children),
- distinction left child/right child,
- nodes are decorated with  $+/-$  signs.

- Very simple combinatorial structure : if  $\tau_n$  counts binary trees with  $n$  leaves, grafting a leaf on an edge shows

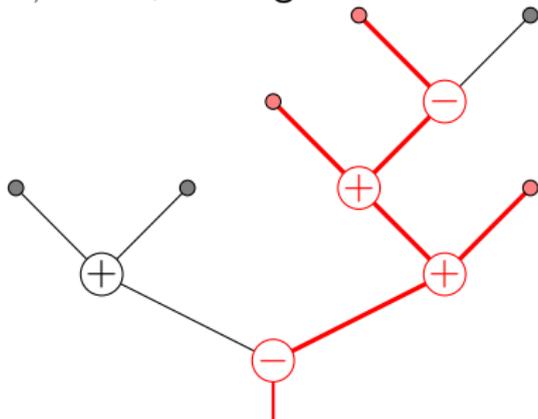
$$(n + 1)\tau_{n+1} = 2(2n - 1)\tau_n$$

so

$$\tau_n = \frac{1}{n} \binom{2n - 2}{n - 1}.$$

# Positive subtrees

- A *positive subtree* is a subset  $A$  of the leaves such that for any  $u, v \in A$ , the highest common ancestor to  $u$  and  $v$  is  $\oplus$ .

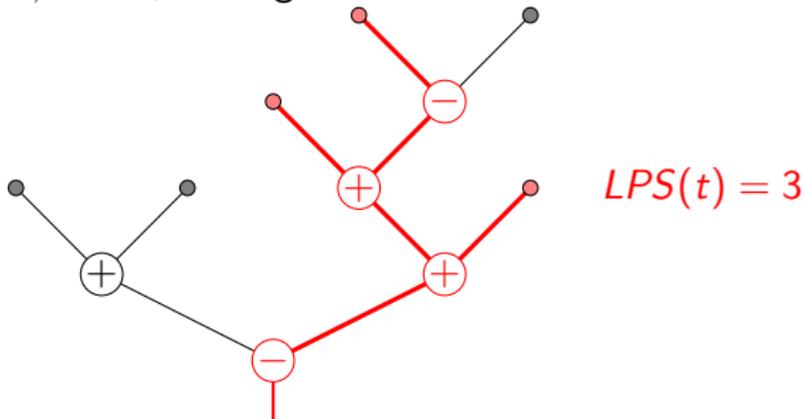


- Largest positive subtree:

$LPS(t) :=$  maximal number of leaves in a positive subtree

# Positive subtrees

- A *positive subtree* is a subset  $A$  of the leaves such that for any  $u, v \in A$ , the highest common ancestor to  $u$  and  $v$  is  $\oplus$ .

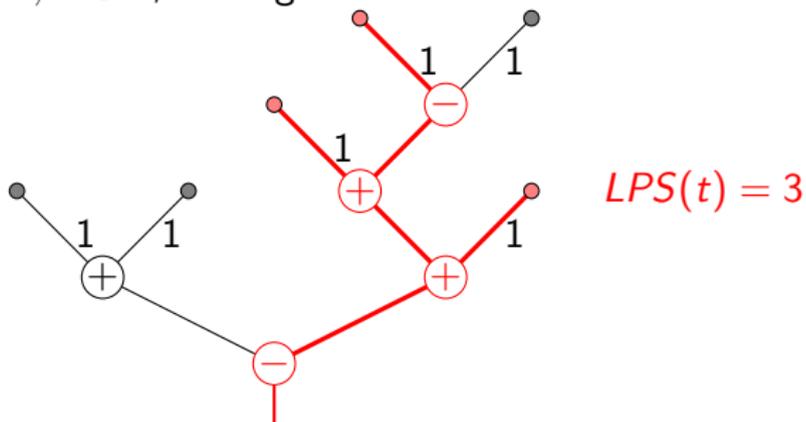


- Largest positive subtree:

$LPS(t) :=$  maximal number of leaves in a positive subtree

# Positive subtrees

- A *positive subtree* is a subset  $A$  of the leaves such that for any  $u, v \in A$ , the highest common ancestor to  $u$  and  $v$  is  $\oplus$ .



- Largest positive subtree:

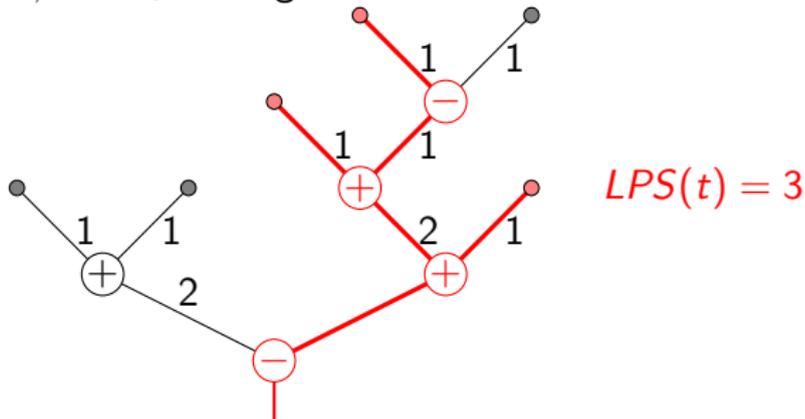
$LPS(t) :=$  maximal number of leaves in a positive subtree

- Easy to compute by a recursive algorithm:  $\oplus$  means "sum" and  $\ominus$  means "max".



# Positive subtrees

- A *positive subtree* is a subset  $A$  of the leaves such that for any  $u, v \in A$ , the highest common ancestor to  $u$  and  $v$  is  $\oplus$ .



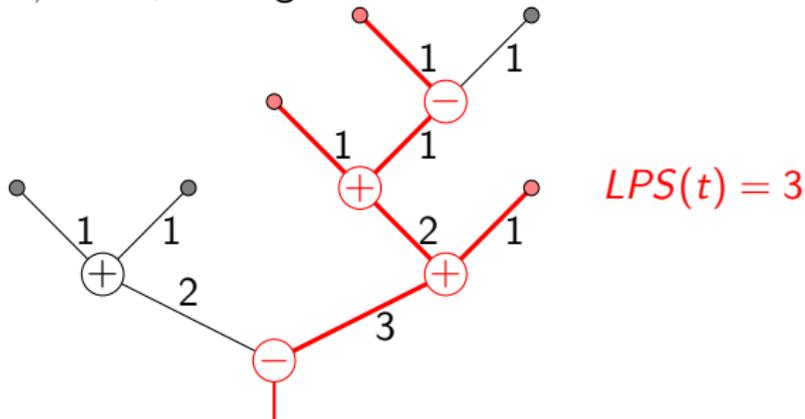
- Largest positive subtree:

$LPS(t) :=$  maximal number of leaves in a positive subtree

- Easy to compute by a recursive algorithm:  $\oplus$  means "sum" and  $\ominus$  means "max".

# Positive subtrees

- A *positive subtree* is a subset  $A$  of the leaves such that for any  $u, v \in A$ , the highest common ancestor to  $u$  and  $v$  is  $\oplus$ .



- Largest positive subtree:

$LPS(t) :=$  maximal number of leaves in a positive subtree

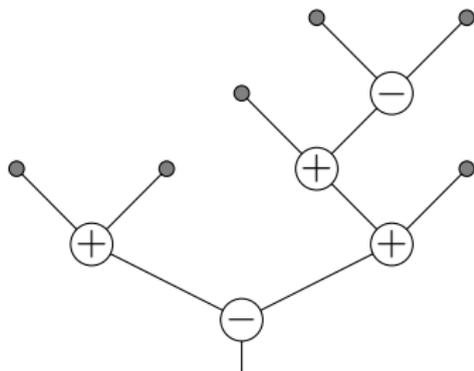
- Easy to compute by a recursive algorithm:  $\oplus$  means "sum" and  $\ominus$  means "max".



- Fix  $p \in [0, 1]$  and  $n$  a positive integer.
- Let  $T_n$  be chosen uniformly at random among binary trees with leaves.
- Signs on the  $n - 1$  nodes are i.i.d. with  $\mathbb{P}(\oplus) = p$  and  $\mathbb{P}(\ominus) = 1 - p$ .
- What is the order of magnitude of  $LPS(T_n)$  as  $n \rightarrow +\infty$ ?

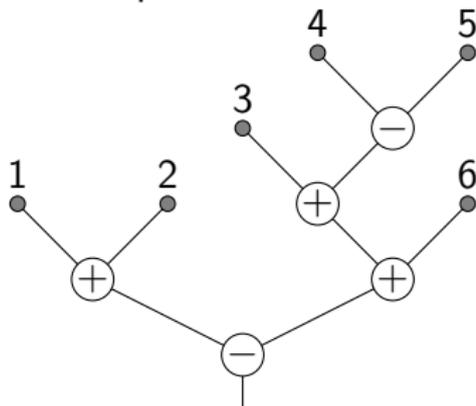
# Motivation: random permutations

- Build a permutation from a signed binary tree:



# Motivation: random permutations

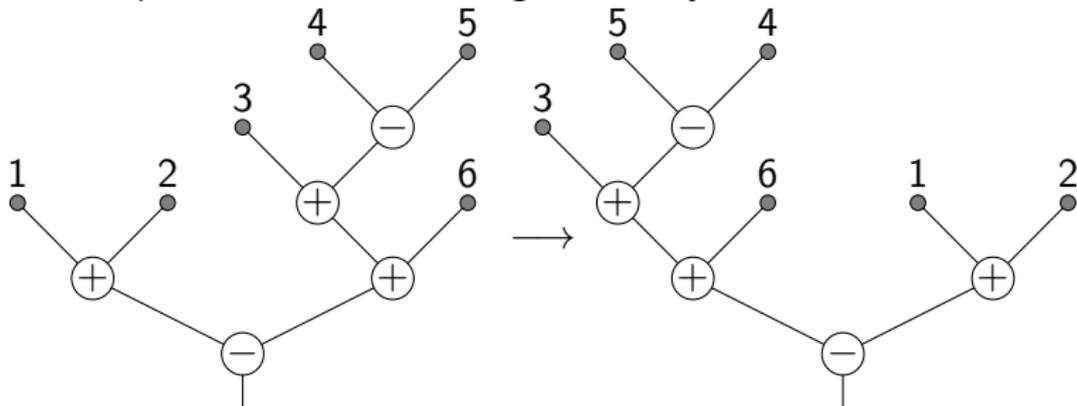
- Build a permutation from a signed binary tree:



- label leaves from left to right,

# Motivation: random permutations

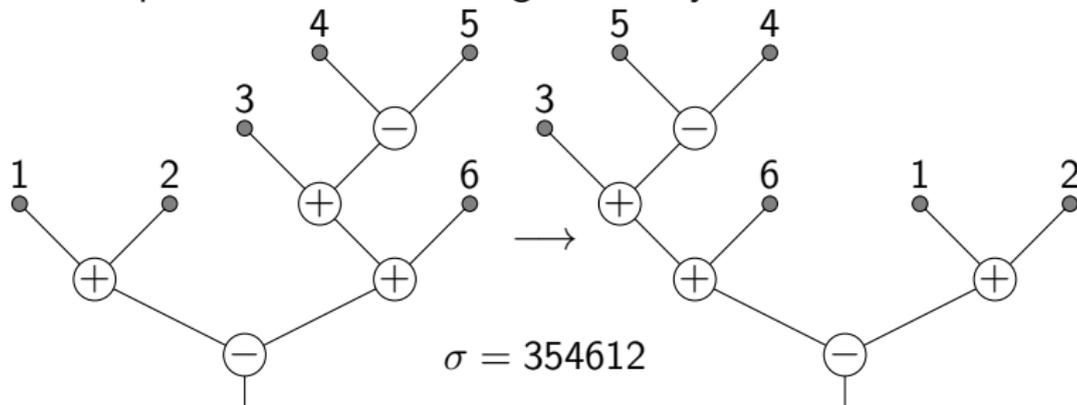
- Build a permutation from a signed binary tree:



- label leaves from left to right,
- swap children at each  $\ominus$  node,

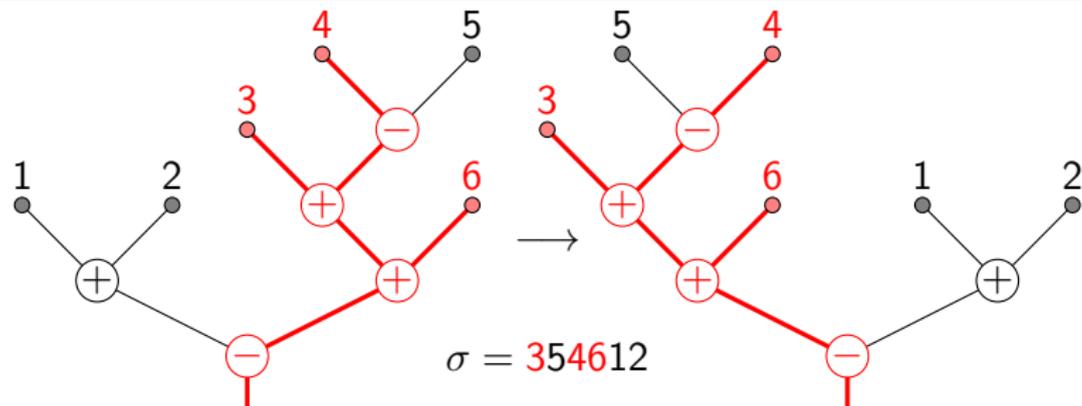
# Motivation: random permutations

- Build a permutation from a signed binary tree:



- label leaves from left to right,
  - swap children at each  $\ominus$  node,
  - read leaves from left to right.
- Permutations that can be obtained in this way are called *separable*.

# Motivation: random permutations



- Equivalent definitions of separable permutations:
  - Permutations avoiding certain patterns,
  - Permutations that can be obtained from  $n$  polynomials by ranking them according to their values at  $0^+$  and  $0^-$ .
- Two indices are swapped in  $\sigma$  if and only if their highest common ancestor is  $\ominus$ , so  $LPS(t)$  corresponds to the largest increasing subsequence of  $\sigma$ .
- The tree  $\rightarrow$  permutation application is not bijective.

# Order of magnitude of $LPS(T_n)$

- Easy bound:  $p\sqrt{n} \lesssim LPS(T_n) \lesssim pn$ .

# Order of magnitude of $LPS(T_n)$

- Easy bound:  $p\sqrt{n} \lesssim LPS(T_n) \lesssim pn$ .
- $LPS(T_n) = o(n)$

[Bassino–Bouvel–Drmota–Féray–Gérim–Maazoun–Pierrot 22].

# Order of magnitude of $LPS(T_n)$

- Easy bound:  $p\sqrt{n} \lesssim LPS(T_n) \lesssim pn$ .
- $LPS(T_n) = o(n)$   
[Bassino–Bouvel–Drmotá–Féray–Gérim–Maazoun–Pierrot 22].
- $n^{\beta_*(p)+o(1)} \leq LPS(T_n) \leq n^{\beta^*(p)+o(1)}$  with  
 $\frac{1}{2} < \beta_*(p) < \beta^*(p) < 1$  [Borga–da Silva–Gwynne 24].

# Order of magnitude of $LPS(T_n)$

- Easy bound:  $p\sqrt{n} \lesssim LPS(T_n) \lesssim pn$ .
- $LPS(T_n) = o(n)$   
[Bassino–Bouvel–Drmotá–Féray–Gérim–Maazoun–Pierrot 22].
- $n^{\beta_*(p)+o(1)} \leq LPS(T_n) \leq n^{\beta^*(p)+o(1)}$  with  
 $\frac{1}{2} < \beta_*(p) < \beta^*(p) < 1$  [Borga–da Silva–Gwynne 24].

## Theorem (Adhikari–Borga–B.–da Silva–Sénizergues 25)

We have the convergence in distribution

$$\frac{LPS(T_n)}{n^{\alpha(p)}} \xrightarrow[n \rightarrow +\infty]{(d)} X(p),$$

where  $X(p) \in (0, +\infty)$  almost surely and

$\alpha(p) = \frac{1}{2(\gamma(p)-1)} \in (\frac{1}{2}, 1)$ , where  $\gamma(p) \in (\frac{3}{2}, 2)$  is the solution to

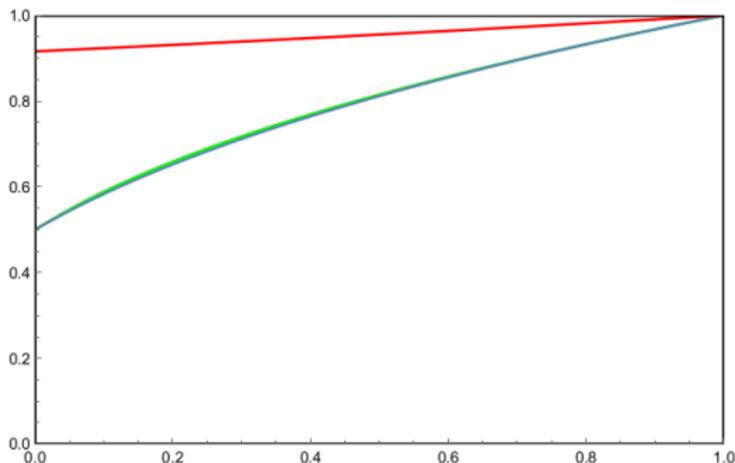
$$-2^{2\gamma-1} + \gamma \int_0^{1/2} x^{1-\gamma} (1-x)^{-1-\gamma} dx = \frac{1-p}{p}.$$

# The exponent $\alpha(p)$

- The equation on  $\alpha$  can be rewritten

$$\frac{1}{2^{1/\alpha} \sqrt{\pi}} \frac{\Gamma\left(\frac{1}{2} - \frac{1}{2\alpha}\right)}{\Gamma\left(1 - \frac{1}{2\alpha}\right)} = \frac{p}{p-1}.$$

- Dependence in  $p$ :  $\beta_*(p) < \alpha(p) < \beta^*(p)$



- Fun fact:  $\max_{p \in [0,1]} (\alpha(p) - \beta_*(p)) < \frac{1}{100}$ .